

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) An arbiter which arbitrates between a plurality of clients generating requests for access to a resource in a computing environment, comprising:

a memory, comprising for each of the plurality of clients:

a request register, which is adapted to record a number of the ~~respective~~ client's access requests; and

a next-client pointer, which is adapted to record an identification of another one of the clients making a subsequent request to access the resource, so as to form a linked list of the requests; and

logic circuitry which is adapted to update the linked list in response to the number of access requests of each of one or more of the clients, and which is adapted to decide, responsive to the linked list, which of the plurality of clients is given access to the resource.

2. (original) An arbiter according to claim 1, wherein the memory comprises at least one list-terminating pointer which indicates an end of the linked list.

3. (original) An arbiter according to claim 2, wherein the at least one list-terminating pointer comprises a tail pointer which indicates a last client in the linked list.

4. (original) An arbiter according to claim 2, wherein the at least one list-terminating pointer comprises a head pointer which indicates a first client in the linked list, and wherein the logic circuitry is operative to decide, responsive to the head pointer, which of the plurality of clients is given access to the resource.

5. (currently amended) An arbiter according to claim 1, wherein the logic circuitry is operative to check whether a client requesting access to the resource has a pending access request, and to update ~~a record of~~ the number of ~~pending~~ access

requests of the client requesting access ~~recorded in the respective register~~ responsive to the check.

6. (original) An arbiter according to claim 1, wherein the logic circuitry is operative to check whether the resource is available, and to allocate the resource responsive to the check.

7. (currently amended) An arbiter according to claim 1, and comprising at least one buffer wherein requests from a specific client are stored before being the number of the requests of the specific client are recorded in the ~~respective~~ request register of the specific client.

8. (original) An arbiter according to claim 1, and comprising a first-in first-out memory wherein requests from the plurality of clients are stored before being transferred sequentially to the memory and the logic circuitry.

9. (original) An arbiter according to claim 1, wherein the memory comprises:

for at least some of the clients, a priority flag which is adapted to record a priority for access to the resource for the at least some clients; and

at least one list-terminating pointer for the priority, which indicates an end of the linked list for the at least some clients.

10. (original) An arbiter according to claim 9, wherein the logic circuitry is adapted to decide, responsive to the linked list and the priority flag, which of the clients is given access to the resource.

11. (currently amended) An arbiter according to claim 1, wherein the logic circuitry is of a size that is ~~substantially~~ independent of ~~the number~~ a quantity of clients served by the arbiter, and wherein the circuitry is adapted to decide, responsive to the numbers of recorded requests, which of the plurality of clients is given access to the resource.

12. (original) An arbiter according to claim 1, wherein a size of the memory scales as a product of the number of clients and a logarithm of the number of clients.

13. (currently amended) An arbiter serving a plurality of clients that generate

requests for access to a resource in a computing environment, comprising:

a memory, comprising a respective register assigned to each of the plurality of clients, which register is adapted to record a number of the respective client's access requests; and

logic circuitry, of a size that is ~~substantially~~ independent of ~~the number a~~ quantity of clients served by the arbiter, which circuitry is adapted to decide, responsive to the number of the recorded requests, which of the plurality of clients is given access to the resource.

14. (currently amended) An arbiter according to claim 13, wherein the size of the memory scales as a product of the ~~number~~ quantity of clients and a logarithm of the ~~number~~ quantity of clients.

15. (currently amended) A method for arbitrating between a plurality of clients generating requests for access to a resource in a computing environment, comprising:

for each of the plurality of clients, recording a number of the client's access requests in a respective, dedicated memory register;

recording for each of the clients, responsive to the requests, a next-client pointer to a subsequent one of the clients requesting the resource, so as to form a linked list of the clients;

updating the linked list in response to the number of access requests of each of one or more of the clients; and

giving the clients access to the resource responsive to the linked list.

16. (original) A method according to claim 15, wherein recording the next-client pointer comprises pointing to an end of the linked list with a list-terminating pointer.

17. (original) A method according to claim 16, wherein pointing to the end of the linked list comprises pointing to a last client in the linked list with a tail pointer.

18. (original) A method according to claim 16, wherein pointing to the end of the linked list comprises pointing to a first client in the linked list with a head pointer, and wherein giving the clients access to the resource comprises giving the clients

access to the resource responsive to the head pointer.

19. (currently amended) A method according to claim 15, wherein recording the number of the client's access requests comprises checking if a client requesting access to the resource has a pending request, and updating the number in the memory register of the client requesting access responsive to the check.

20. (original) A method according to claim 15, wherein giving the clients access comprises checking whether the resource is available, and allocating the resource responsive to the check.

21. (currently amended) A method according to claim 15, wherein recording the number of the client's access requests comprises providing at least one buffer and storing the requests from the client in the buffer prior to recording the number of the client's access requests.

22. (original) A method according to claim 15, wherein giving the clients access comprises assigning priorities to at least some of the clients, and forming a linked list of the prioritized clients.

23. (original) A method according to claim 22, wherein giving the clients access to the resource comprises deciding, responsive to the assigned priorities and the linked list, which of the clients is given access to the resource.

24. (currently amended) A method for arbitrating between a plurality of clients generating requests for access to a resource in a computing environment, comprising:

for each of the plurality of clients, recording a number of the client's access requests;

providing logic circuitry of a size that is ~~substantially~~ independent of ~~the number~~ a quantity of the plurality of clients; and

utilizing the logic circuitry to decide, responsive to the number of recorded requests, which of the plurality of clients is given access to the resource.